Overview of Multirate Signal Processing

Digital transformation of the sampling rate of signals, or signal processing with different sampling rates in the system.

Applications

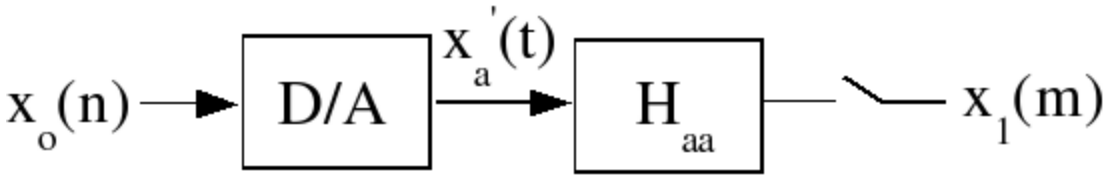
1. **Sampling-rate conversion** CD to DAT format change, for example.
2. **Improved D/A, A/D conversion** oversampling converters; which reduce performance requirements on anti-aliasing or reconstruction filters
3. **FDM channel modulation and processing** bandwidth of individual channels is much less than the overall bandwidth
4. **Subband coding of speech and images** Eyes and ears are not as sensitive to errors in higher frequency bands, so many coding schemes split signals into different frequency bands and quantize higher-frequency bands with much less precision.

Outline of Multirate DSP material

1. General Rate-changing System
2. Integer-factor Interpolation and Decimation and Rational-factor Rate Changing
3. Efficient Multirate Filter Structures
4. Optimal Filter Design for Multirate Systems
5. Multi-stage Multirate Systems
6. Oversampling D/As
7. Perfect-Reconstruction Filter Banks and Quadrature Mirror Filters

General Rate-Changing Procedure

This procedure is motivated by an analog-based method: one conceptually simple method to change the sampling rate is to simply convert a digital signal to an analog signal and resample it! ([link])



$$H_{\text{aa}}(\Omega) = \begin{cases} 1 & \text{if} \quad |\Omega| < \frac{\pi}{T_1} \\ 0 & \text{otherwise} \end{cases}$$

$$h_{\text{aa}}(t) = \frac{\sin\left(\frac{\pi}{T_1}t\right)}{\frac{\pi}{T_1}t}$$

Recall the ideal D/A:
Equation:

$$x'_a(t) = \sum_{n=-\infty}^{\infty} x_0(n) \frac{\sin\left(\frac{\pi(t-nT_0)}{T_0}\right)}{\frac{\pi(t-nT_0)}{T_0}}$$

The problems with this scheme are:

1. A/D, D/A, filters cost money
2. imperfections in these devices introduce errors

Digital implementation of rate-changing according to this formula has three problems:

1. Infinite sum: The solution is to truncate. Consider sinc $(t) \simeq 0$ for $t < t_1, t > t_2$: Then $mT_1 - nT_0 < t_1$ and $mT_1 - nT_0 > t_2$ which implies

$$N_1 = \left\lceil \frac{mT_1 - t_2}{T_0} \right\rceil$$

$$N_2 = \left\lfloor \frac{mT_1 - t_1}{T_0} \right\rfloor$$

$$x_1(m) = \sum_{n=N_1}^{N_2} x_0(n)\ \mathrm{sinc}_{T'}\ (mT_1 - nT_0)$$

> **Note:** This is essentially lowpass filter design using a boxcar window: other finite-length filter design methods could be used for this.

2. Lack of causality: The solution is to delay by $\max\{|N|\}$ samples. The mathematics of the analog portions of this system can be implemented digitally.
   Equation:

$$\begin{aligned} x_1(m) &= h_{\mathrm{aa}}(t) * x'_a(t)|_{t=mT_1} \\ &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_0(n) \frac{\sin\left(\frac{\pi(mT_1-\tau-nT_0)}{T_0}\right)}{\frac{\pi(mT_1-\tau-nT_0)}{T_0}} \frac{\sin\left(\frac{\pi\tau}{T_1}\right)}{\frac{\pi\tau}{T_1}}\ \mathrm{d}\,\tau \end{aligned}$$

   Equation:

$$\begin{aligned} x_1(m) &= \sum_{n=-\infty}^{\infty} x_0(n) \frac{\sin\left(\frac{\pi}{T'}(mT_1-nT_0)\right)}{\frac{\pi}{T'}(mT_1-nT_0)} \Bigg|_{T'=\max\{T_0,T_1\}} \\ &= \sum_{n=-\infty}^{\infty} x_0(n)\ \mathrm{sinc}_{T'}\ (mT_1 - nT_0) \end{aligned}$$

   So we have an all-digital formula for **exact** digital-to-digital rate changing!
3. Cost of computing $\mathrm{sinc}_{T'}\ (mT_1 - nT_0)$: The solution is to precompute the table of $\mathrm{sinc}\,(t)$ values. However, if $\frac{T_1}{T_0}$ is not a rational fraction, an infinite number of samples will be needed, so some approximation will have to be tolerated.

Note: Rate transformation of any rate to any other rate can be accomplished digitally with arbitrary precision (if some delay is acceptable). This method is used in practice in many cases. We will examine a number of special cases and computational improvements, but in some sense everything that follows are details; the above idea is the central idea in multirate signal processing.

Useful references for the traditional material (everything except PRFBs) are Crochiere and Rabiner, 1981 and Crochiere and Rabiner, 1983. A more recent tutorial is Vaidyanathan; see also Rioul and Vetterli. References to most of the original papers can be found in these tutorials.

Interpolation, Decimation, and Rate Changing by Integer Fractions

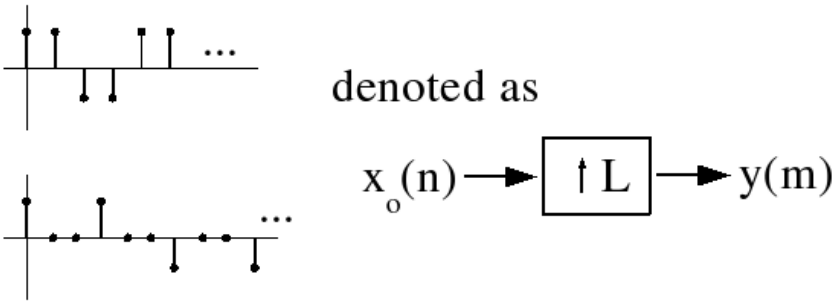
Interpolation: by an integer factor L

Interpolation means increasing the sampling rate, or filling in in-between samples. Equivalent to sampling a bandlimited analog signal L times faster. For the ideal interpolator,
Equation:

$$X_1(\omega) = \begin{cases} X_0(L\omega) \text{ if } |\omega| < \frac{\pi}{L} \\ 0 \text{ if } \frac{\pi}{L} \leq |\omega| \leq \pi \end{cases}$$

We wish to accomplish this digitally. Consider [link] and [link].
Equation:

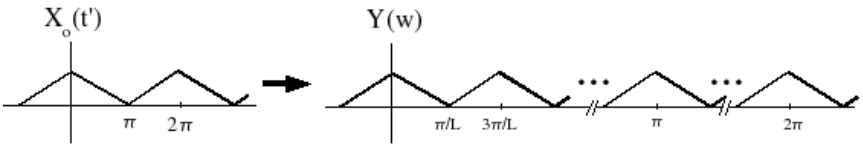
$$y(m) = \begin{cases} X_0\left(\frac{m}{L}\right) \text{ if } m = \{0, \pm(L), \pm(2L), \ldots\} \\ 0 \text{ otherwise} \end{cases}$$



The DTFT of $y(m)$ is
Equation:

$$\begin{aligned} Y(\omega) &= \sum_{m=-\omega}^{\infty} y(m) e^{-(i\omega m)} \\ &= \sum_{n=-\infty}^{\infty} x_0(n) e^{-(i\omega L n)} \\ &= \sum_{n=-\infty}^{\infty} x(n) e^{-(i\omega L n)} \\ &= X_0(\omega L) \end{aligned}$$

Since $X_0(\omega')$ is periodic with a period of 2π, $X_0(L\omega) = Y(\omega)$ is periodic with a period of $\frac{2\pi}{L}$ (see [link]).



By inserting zero samples between the samples of $x_0(n)$, we obtain a signal with a scaled frequency response that simply replicates $X_0(\omega')$ L times over a 2π interval!

Obviously, the desired $x_1(m)$ can be obtained simply by lowpass filtering $y(m)$ to remove the replicas.

Equation:

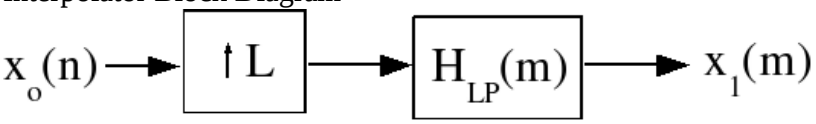
$$x_1(m) = y(m)*h_L(m)$$

Given

$$H_L(m) = \begin{cases} 1 & \text{if } |\omega| < \frac{\pi}{L} \\ 0 & \text{if } \frac{\pi}{L} \le |\omega| \le \pi \end{cases}$$

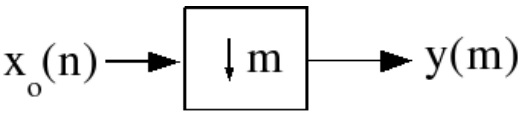
In practice, a finite-length lowpass filter is designed using any of the methods studied so far ([link]).

Interpolator Block Diagram

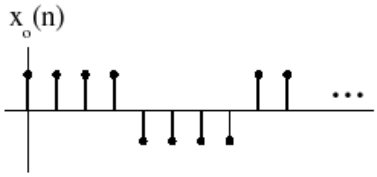


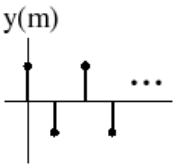
Decimation: sampling rate reduction (by an integer factor M)

Let $y(m) = x_0(Lm)$ ([link])



That is, keep only every Lth sample ([link])





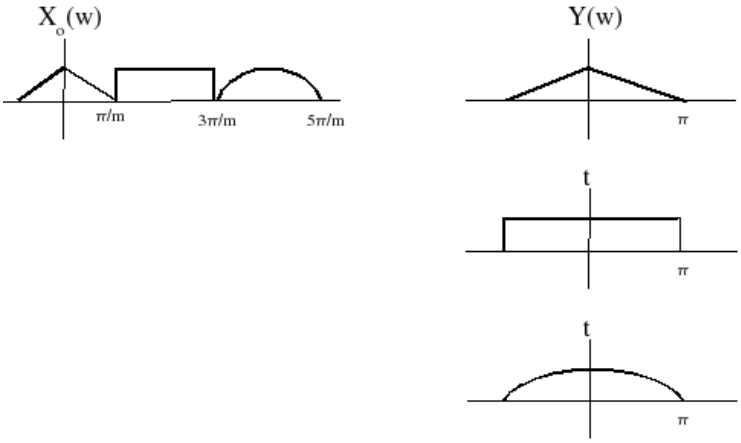
In frequency (DTFT):
Equation:

$$\begin{aligned} Y(\omega) &= \sum_{m=-\infty}^{\infty} y(m) e^{-(i\omega m)} \\ &= \sum_{m=-\infty}^{\infty} x_0(Mm) e^{-(i\omega m)} \\ &= \sum_{n=-\infty}^{\infty} x_0(n) \sum_{k=-\infty}^{\infty} \delta(n-Mk) e^{-\left(i\omega \frac{n}{M}\right)}\Big|_{n=Mm} \\ &= \sum_{n=-\infty}^{\infty} x_0(n) \sum_{k=-\infty}^{\infty} \delta(n-Mk) e^{-(i\omega' n)}\Big|_{\omega'=\frac{\omega}{M}} \\ &= \mathrm{DTFT}\left[x_0(n)\right] * \mathrm{DTFT}\left[\sum \delta(n-Mk)\right] \end{aligned}$$

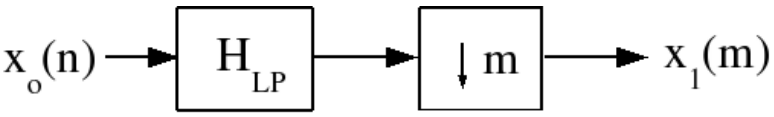
Now $\mathrm{DTFT}\left[\sum \delta(n-Mk)\right] = 2\pi \sum_{k=0}^{M-1} X(k)\delta\left(\omega\prime - \frac{2\pi k}{M}\right)$ for $|\omega| < \pi$ as shown in homework #1 , where $X(k)$ is the DFT of one period of the periodic sequence. In this case, $X(k) = 1$ for $k \in \{0, 1, \ldots, M-1\}$ and $\mathrm{DTFT}\left[\sum \delta(n-Mk)\right] = 2\pi \sum_{k=0}^{M-1} \delta\left(\omega\prime - \frac{2\pi k}{M}\right)$.
Equation:

$$\begin{aligned} \mathrm{DTFT}\left[x_0(n)\right] * \mathrm{DTFT}\left[\sum \delta(n-Mk)\right] &= X_0(\omega') * 2\pi \sum_{k=0}^{M-1} \delta\left(\omega\prime - \frac{2\pi k}{M}\right) \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} X_0(\mu') \left(2\pi \sum_{k=0}^{M-1} \delta\left(\omega\prime - \mu\prime - \frac{2\pi k}{M}\right)\right) \mathrm{d}\,\mu' \\ &= \sum_{k=0}^{M-1} X_0\left(\omega\prime - \frac{2\pi k}{M}\right) \end{aligned}$$

so $Y(\omega) = \sum_{k=0}^{M-1} X_0\left(\frac{\omega}{M} - \frac{2\pi k}{M}\right)$ i.e., we get **digital aliasing**.([link])

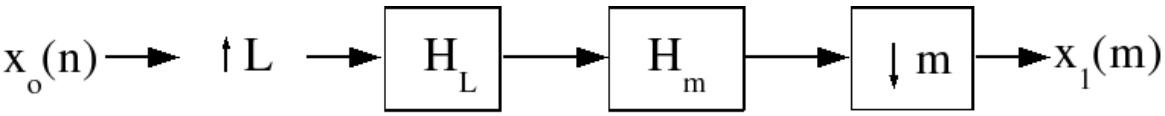


Usually, we prefer not to have aliasing, so the downsampler is preceded by a lowpass filter to remove all frequency components above $|\omega| < \frac{\pi}{M}$ ([link]).



Rate-Changing by a Rational Fraction L/M

This is easily accomplished by interpolating by a factor of L, then decimating by a factor of M ([link]).



The two lowpass filters can be combined into one LP filter with the lower cutoff,

$$H(\omega) = \begin{cases} 1 & \text{if} \quad |\omega| < \frac{\pi}{\max\{L,M\}} \\ 0 & \text{if} \quad \frac{\pi}{\max\{L,M\}} \leq |\omega| \leq \pi \end{cases}$$

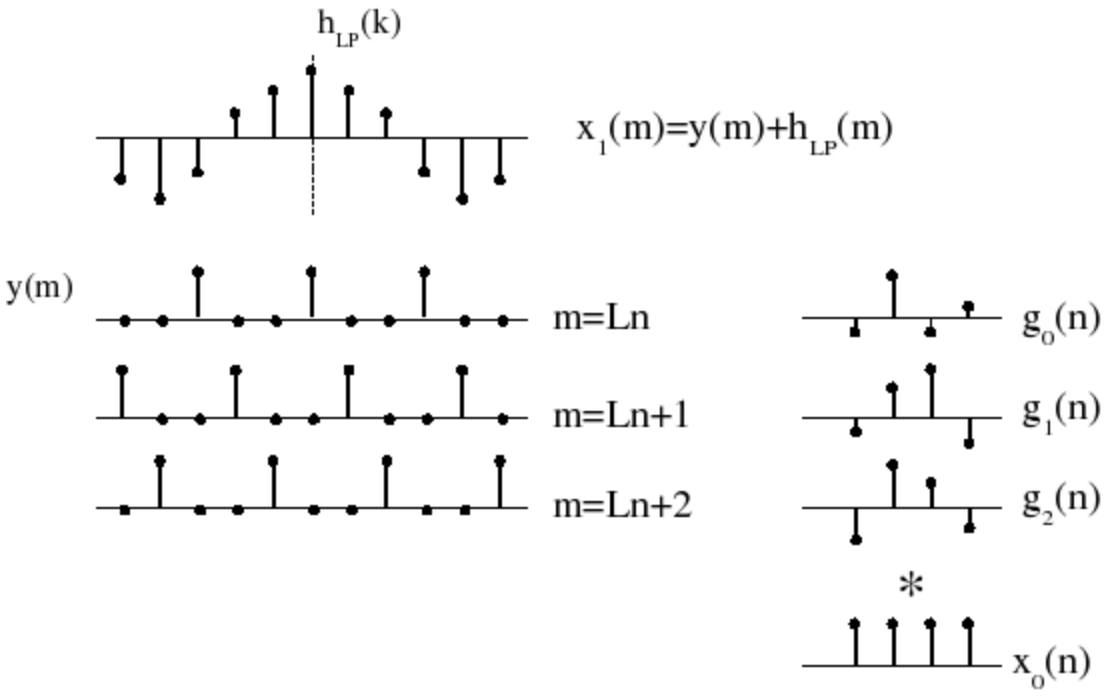
Obviously, the computational complexity and simplicity of implementation will depend on $\frac{L}{M}$: $2/3$ will be easier to implement than $1061/1060$!

Efficient Multirate Filter Structures

Rate-changing appears expensive computationally, since for both decimation and interpolation the lowpass filter is implemented at the higher rate. However, this is not necessary.

Interpolation

For the interpolator, most of the samples in the upsampled signal are zero, and thus require no computation. ([link])

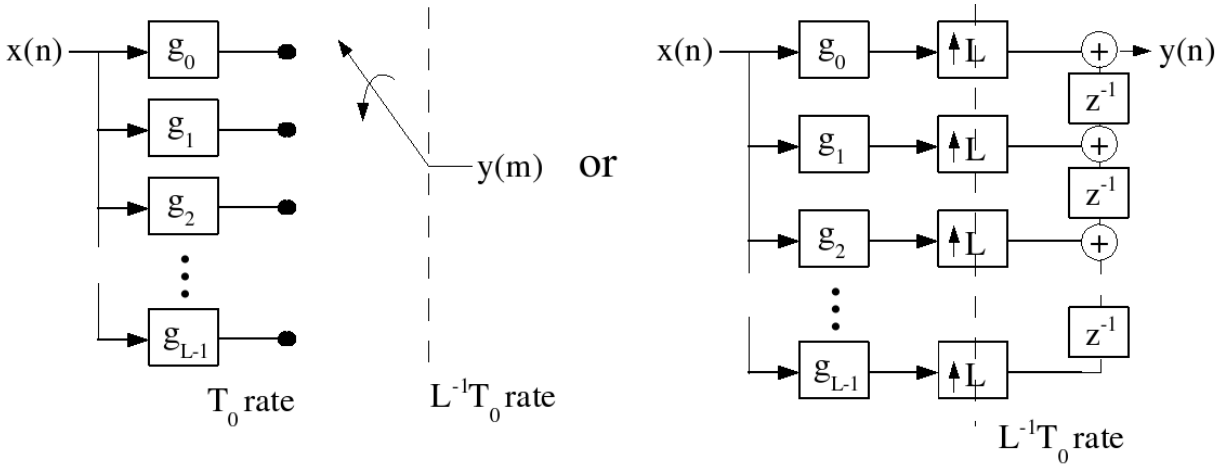


For $m = L\left\lfloor \frac{m}{L} \right\rfloor + m \bmod L$ and $p = m \bmod L$,
Equation:

$$\begin{aligned} x_1(m) &= \sum_{m=N_1}^{N_2} h_{Lp}(m) y(m) \\ &= \sum_{k=\frac{N_1}{L}}^{\frac{N_2}{L}} g_p(k) x_0\left(\left\lfloor \frac{m}{L} \right\rfloor - k\right) \end{aligned}$$

$$g_p(n) = h(Ln + p)$$

Pictorially, this can be represented as in [link].



These are called **polyphase structures**, and the $g_p(n)$ are called **polyphase filters**.

Computational cost
If $h(m)$ is a length-N filter:

- No simplification: $\frac{N}{T_1} = \frac{LN}{T_0} \frac{\text{computations}}{\text{sec}}$
- Polyphase structure: $\left(L \frac{L}{N} \frac{1}{T_0^o}\right) \frac{\text{computations}}{\text{sec}} = \frac{N}{T_0}$ where L is the number of filters, $\frac{N}{L}$ is the taps/filter, and $\frac{1}{T_0}$ is the rate.

Thus we save a factor of L by not being dumb.

Note: For a given precision, N is proportional to L, (why?), so the computational cost does increase with the interpolation rate.

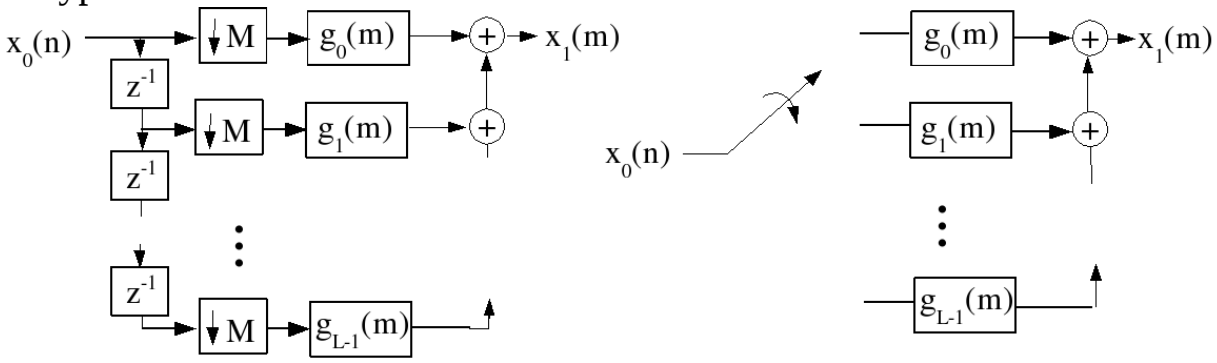
Note: Can similar computational savings be obtained with IIR structures?

Efficient Decimation Structures

We only want every Mth output, so we compute only the outputs of interest. ([link])

$$x_1(m) = \sum_{k=N_1}^{N_2} x_0(Lm-k)h(k)$$

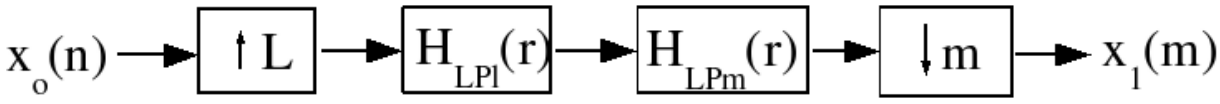
Polyphase Decimation Structure



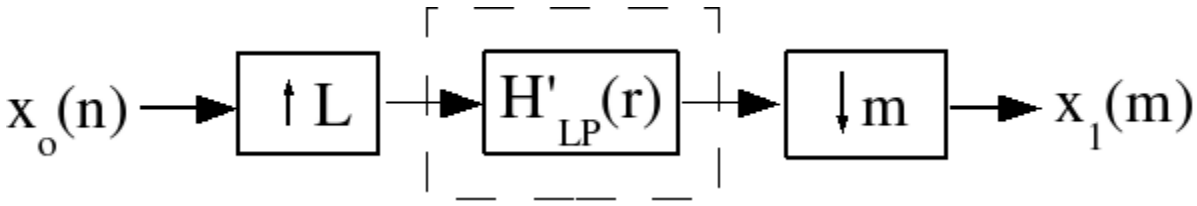
The decimation structures are flow-graph reversals of the interpolation structure. Although direct implementation of the full filter for every Mth sample is obvious and straightforward, these polyphase structures give some idea as to how one might evenly partition the computation over M cycles.

Efficient L/M rate changers

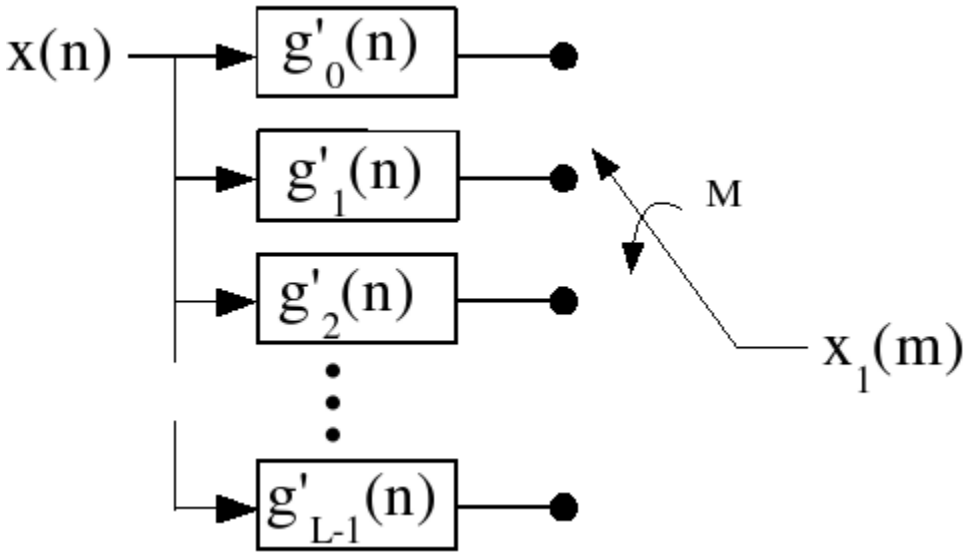
Interpolate by L and decimate by M ([link]).



Combine the lowpass filters ([link]).



We can couple the lowpass filter either to the interpolator or the decimator to implement it efficiently ([link]).



Of course we only compute the polyphase filter output selected by the decimator.

Computational Cost

Every $T_1 = \frac{M}{L} T_0$ seconds, compute one polyphase filter of length $\frac{N}{L}$, or

$$\frac{\frac{N}{L}}{T_1} = \frac{\frac{N}{L}}{\frac{M}{L} T_0} = \frac{N}{M T_0} \frac{\text{multiplies}}{\text{second}}$$

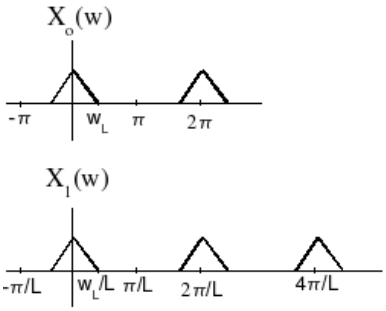
However, note that N is proportional to $\max\{L, M\}$.

Filter Design for Multirate Systems

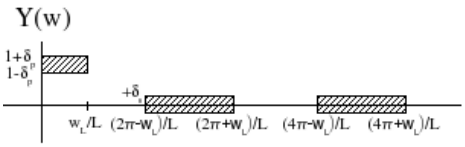
The filter design techniques learned earlier can be applied to the design of filters in multirate systems, with a few twists.

Example:
Design a factor-of-L interpolator for use in a CD player, we might wish that the out-of-band error be below the least significant bit, or 96dB down, and $< 0.05\%$ error in the passband, so these specifications could be used for optimal L_∞ filter design.

In a CD player, the sampling rate is 44.1kHz, corresponding to a Nyquist frequency of 22.05kHz, but the sampled signal is bandlimited to 20kHz. This leaves a small transition band, from 20kHz to 24.1kHz. However, note that in any case where the signal spectrum is zero over some band, this introduces **other** zero bands in the scaled, replicated spectrum ([link]).



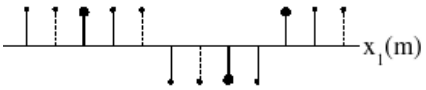
So we need only control the filter response in the stopbands over the frequency regions with nonzero energy. ([link])



The extra "don't care" bands allow a given set of specifications to be satisfied with a shorter-length filter.

Direct polyphase filter design

Note that in an integer-factor interpolator, each set of output samples $x_1(Ln+p)$, $p = \{0, 1, \ldots, L-1\}$, is created by a different polyphase filter $g_p(n)$, which has no interaction with the other polyphase filters except in that they each interpolate the same signal. We can thus treat the design of each polyphase filter **independently**, as an $\frac{N}{L}$-length filter design problem. ([link])



Each $g_p(n)$ produces samples $x_1(Ln+p) = x_0\left(n+\frac{p}{L}\right)$, where $n+\frac{p}{L}$ is not an integer. That is, $g_p(n)$ is to produce an output signal (at a T_0 rate) that is $x_0(n)$ time-advanced by a non-integer advance $\frac{p}{L}$.

The desired response of this polyphase filter is thus

$$H_{Dp}(\omega) = e^{\frac{i\omega p}{L}}$$

for $|\omega| \leq \pi$, an all-pass filter with a linear, non-integer, phase. Each polyphase filter can be designed independently to approximate this response according to any of the design criteria developed so far.
Exercise:

Problem: What should the polyphase filter for $p = 0$ be?

Solution:

A delta function: $h_0(n) = \delta(n')$

Example:
Least-squares Polyphase Filter Design

- **Deterministic x(n)** Minimize

$$\sum_{n=-\infty}^{\infty} \left(|x(n) - x_d(n)|\right)^2$$

Given $x(n) = x(n)^* h(n)$ and $x_d(n) = x(n)^* h_d(n)$. Using Parseval's theorem, this becomes
Equation:

$$\begin{aligned} \min\left\{\sum_{n=-\infty}^{\infty}\left(|x(n)-x_d(n)|\right)^2\right\} &= \min\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi}\left(|X(\omega)H(\omega)-X(\omega)H_d(\omega)|\right)^2 \mathrm{d}\,\omega\right\} \\ &= \min\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi}|H(\omega)-H_d(\omega)|\left(|X(\omega)|\right)^2 \mathrm{d}\,\omega\right\} \end{aligned}$$

This is simply weighted least squares design, with $\left(|X(\omega)|\right)^2$ as the weighting function.
- **stochastic X(ω)**
Equation:

$$\begin{aligned} \min\left\{E\left[\left(|x(n)-x_d(n)|\right)^2\right]\right\} &= E\left[\left(|x(n)^*(h(n)-h_d(n))|\right)^2\right] \\ &= \min\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi}\left(|H_d(\omega)-H(\omega)|\right)^2 S_{xx}(\omega)\,\mathrm{d}\,\omega\right\} \end{aligned}$$

$S_{xx}(\omega)$ is the power spectral density of x.

$$S_{xx}(\omega) = \mathrm{DTFT}\left[r_{xx}(k)\right]$$

$$r_{xx}(k) = E\Big[x(k+l)x(l)\Big]$$

Again, a weighted least squares filter design problem.

Exercise:

Problem: Is it feasible to use IIR polyphase filters?

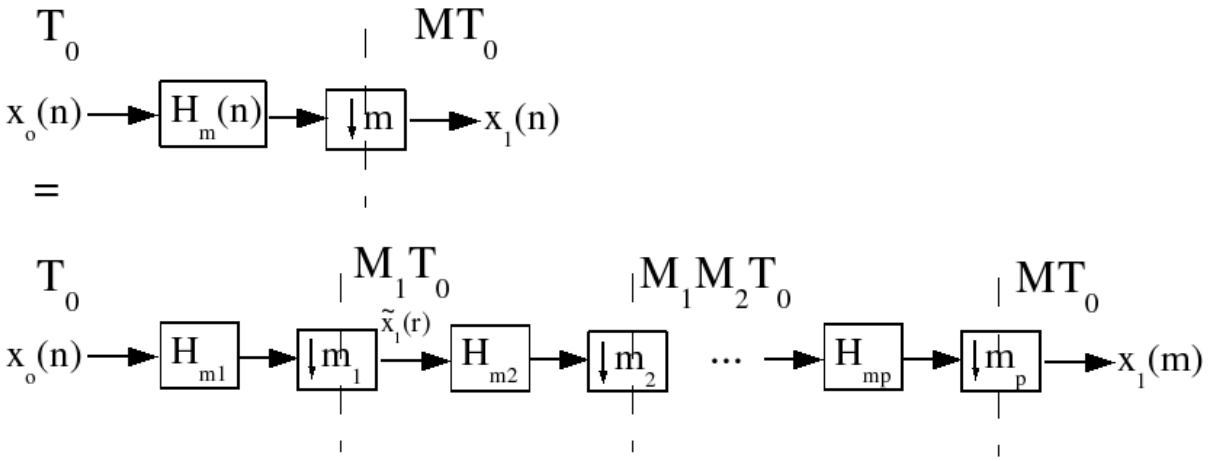
Solution:

The recursive feedback of previous outputs means that portions of each IIR polyphase filter must be computed for every input sample; this usually makes IIR filters more expensive than FIR implementations.

Multistage Multirate Systems

Multistage multirate systems are often more efficient. Suppose one wishes to decimate a signal by an integer factor M, where M is a composite integer $M = M_1 M_2 \ldots M_p = \prod_{i=1}^{p} M_i$. A decimator can be implemented in a multistage fashion by first decimating by a factor M_1 , then decimating this signal by a factor M_2 , etc. ([link])

Multistage decimator



Multistage implementations are of significant practical interest only if they offer significant computational savings. In fact, they often do!

The computational cost of a **single-stage** interpolator is:

$$\frac{N}{MT_0}\frac{\text{taps}}{\text{sec}}$$

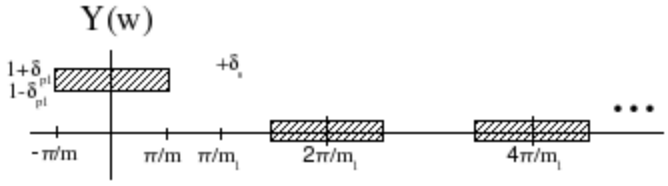
The computational cost of a **multistage** interpolator is:

$$\frac{N_1}{M_1 T_0} + \frac{N_2}{M_1 M_2 T_0} + \ldots + \frac{N_p}{MT_0}$$

The first term is the most significant, since the rate is highest. Since $N_i \propto M_i$ for a lowpass filter, it is not immediately clear that a multistage system should require less computation. However, the multistage structure relaxes the requirements on the filters, which reduces their length and makes the overall computation less.

Filter design for Multi-stage Structures

Ostensibly, the first-stage filter must be a lowpass filter with a cutoff at $\frac{\pi}{M_1}$, to prevent aliasing after the downsampler. However, note that aliasing outside the **final overall** passband $|\omega| < \frac{\pi}{M}$ is of no concern, since it will be removed by later stages. We only need prevent aliasing into the band $|\omega| < \frac{\pi}{M}$; thus we need only specify the passband over the interval $|\omega| < \frac{\pi}{M}$, and the stopband over the intervals $\omega \in \left[\frac{2\pi k}{M_1} - \frac{\pi}{M}, \frac{2\pi k}{M_1} + \frac{\pi}{M}\right]$, for $k \in \{1, \ldots, M-1\}$. ([link])



Of course, we don't want **gain** in the transition bands, since this would need to be suppressed later, but otherwise we don't care about the response in those regions. Since the transition bands are so large, the required filter turns out to be quite short. The final stage has no "don't care" regions; however, it is operating at a low rate, so it is relatively unimportant if the final filter turns out to be rather long!

L-infinity Tolerances on the Pass and Stopbands

The overall response is a cascade of multiple filters, so the worst-case overall passband deviation, assuming all the peaks just happen to line up, is

$$1 + \delta_{p_{\text{ov}}} = \prod_{i=1}^{p} 1 + \delta_{p_i}$$

$$1 - \delta_{p_{\text{ov}}} = \prod_{i=1}^{p} 1 - \delta_{p_i}$$

So one could choose all filters to have equal specifications and require for each-stage filter. For $\delta_{p_{\text{ov}}} \ll 1$,

$$1 + \delta_{p_i}^{+} \leq \sqrt[p]{1 + \delta_{p_{\text{ov}}}} \simeq 1 + p^{-1}\delta_{p_{\text{ov}}}$$

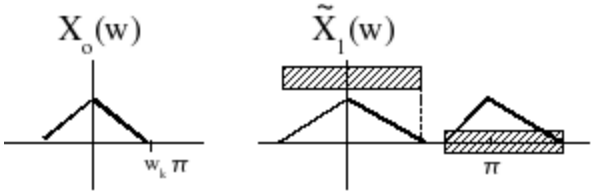
$$1 - \delta_{p_i}^{-} \leq \sqrt[p]{1 - \delta_{p_{\text{ov}}}} \simeq 1 - p^{-1}\delta_{p_{\text{ov}}}$$

Alternatively, one can design later stages (at lower computation rates) to compensate for the passband ripple in earlier stages to achieve exceptionally accurate passband response.

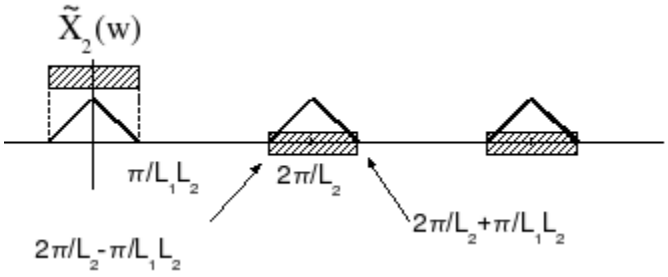
δ_s remains essentially unchanged, since the worst-case scenario is for the error to alias into the passband and undergo no further suppression in subsequent stages.

Interpolation

Interpolation is the flow-graph reversal of the multi-stage decimator. The first stage has a cutoff at $\frac{\pi}{L}$ ([link]):



However, all subsequent stages have large bands without signal energy, due to the earlier stages ([link]):

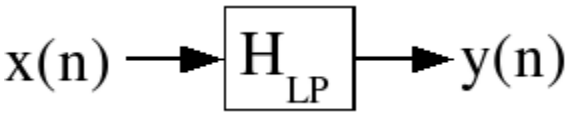


The order of the filters is reversed, but otherwise the filters are identical to the decimator filters.

Efficient Narrowband Lowpass Filtering

A very narrow lowpass filter requires a very long FIR filter to achieve reasonable resolution in the frequency response. However, were the input sampled at a lower rate, the cutoff frequency would be correspondingly higher, and the filter could be shorter!

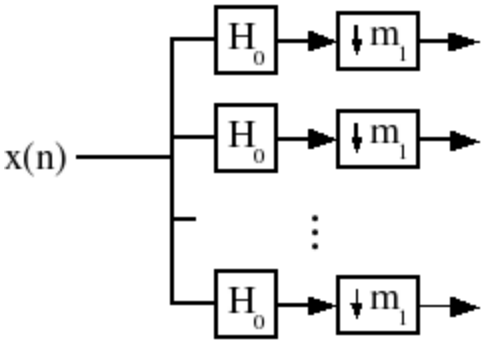
The transition band is also broader, which helps as well. Thus, [link] can be implemented as [link].



and in practice the inner lowpass filter can be coupled to the decimator or interpolator filters. If the decimator and interpolator are implemented as multistage structures, the overall algorithm can be dramatically more efficient than direct implementation!

DFT-Based Filterbanks

One common application of multirate processing arises in multirate, multi-channel filter banks ([[link]]).



One application is separating frequency-division-multiplexed channels. If the filters are narrowband, the output channels can be decimated without significant aliasing.

Such structures are especially attractive when they can be implemented efficiently. For example, if the filters are simply frequency modulated (by $e^{-\left(i\frac{2\pi k}{L}n\right)}$) versions of each other, they can be efficiently implemented using FFTs!

Furthermore, there are classes of filters called **perfect reconstruction filters** which are of finite length but from which the signal can be reconstructed exactly (using all M channels), even though the output of each channel experiences aliasing in the decimation step. These types of filterbanks have received a lot of research attention, culminating in wavelet theory and techniques.

Uniform DFT Filter Banks

Suppose we wish to split a digital input signal into N frequency bands, uniformly spaced at center frequencies $\omega_k = \frac{2\pi k}{N}$, for $0 \leq k \leq N-1$. Consider also a lowpass filter $h(n)$, $H(\omega) \simeq \begin{cases} 1 & \text{if} \quad |\omega| < \frac{\pi}{N} \\ 0 & \text{otherwise} \end{cases}$. Bandpass filters can be constructed which have the frequency response

$$H_k(\omega) = H\left(\omega + \frac{2\pi k}{N}\right)$$

from

$$h_k(n) = h(n)e^{-\left(i\frac{2\pi kn}{N}\right)}$$

The output of the *k*th bandpass filter is simply (assume $h(n)$ are FIR)
Equation:

$$\begin{aligned} x(n)*h_k(n) &= \sum_{m=0}^{M-1} x(n-m)h(m)e^{-\left(i\frac{2\pi km}{N}\right)} \\ &= y_k(n) \end{aligned}$$

This looks suspiciously like a DFT, except that $M \neq N$, in general. However, if we fix $M = N$, then we can compute **all** $y_k(n)$ outputs simultaneously using an FFT of $x(n-m)h(m)$: The
$\text{kth FFT frequency output} = y_k(n)$! So the cost of computing all of these filter banks outputs is $O[N \log N]$, rather than N^2, per a given n. This is very useful for efficient implementation of **transmultiplexors** (FDM to TDM).
Exercise:

Problem:

How would we implement this efficiently if we wanted to decimate the individual channels $y_k(n)$ by a factor of N, to their approximate Nyquist bandwidth?

Solution:

Simply step by N time samples between FFTs.

Exercise:

Problem:

Do you expect significant aliasing? If so, how do you propose to combat it? Efficiently?

Solution:

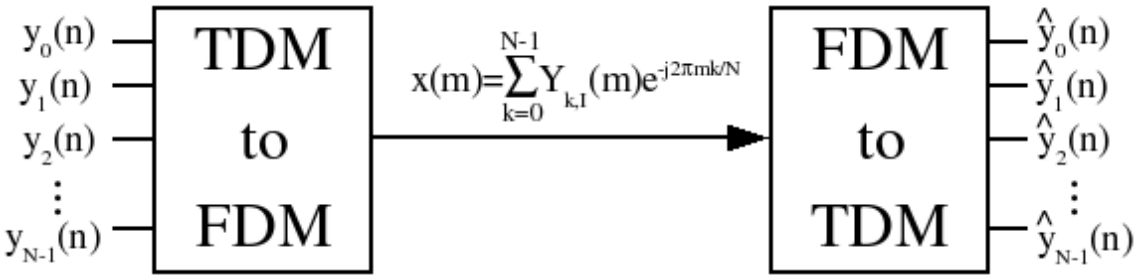
Aliasing should be expected. There are two ways to reduce it:

1. Decimate by less ("oversample" the individual channels) such as decimating by a factor of $\frac{N}{2}$. This is efficiently done by time-stepping by the appropriate factor.
2. Design better (and thus longer) filters, say of length LN. These can be efficiently computed by producing only N (every Lth) FFT outputs using simplified FFTs.

Exercise:

Problem:

How might one convert from N input channels into an FDM signal efficiently? ([link])



Note: Such systems are used throughout the telephone system, satellite communication links, etc.

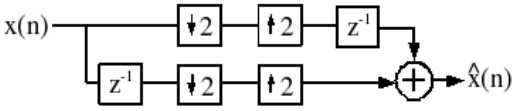
Solution:

Use an FFT and an inverse FFT for the modulation (TDM to FDM) and demodulation (FDM to TDM), respectively.

Quadrature Mirror Filterbanks (QMF)

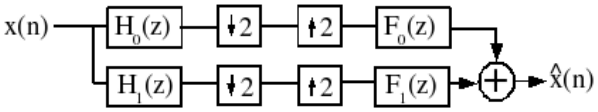
Although the DFT filterbanks are widely used, there is a problem with aliasing in the decimated channels. At first glance, one might think that this is an insurmountable problem and must simply be accepted. Clearly, with FIR filters and maximal decimation, aliasing will occur. However, a simple example will show that it is possible to **exactly cancel out aliasing** under certain conditions!!!

Consider the following trivial filterbank system, with two channels. ([link])



Note $\widehat{x}(n) = x(n)$ with no error whatsoever, although clearly aliasing occurs in both channels! Note that the overall data rate is still the Nyquist rate, so there are clearly enough degrees of freedom available to reconstruct the data, if the filterbank is designed carefully. However, this isn't splitting the data into separate frequency bands, so one questions whether something other than this trivial example could work.

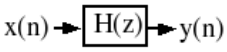
Let's consider a general two-channel filterbank, and try to determine conditions under which aliasing can be cancelled, and the signal can be reconstructed perfectly ([link]).



Let's derive $\widehat{x}(n)$, using z-transforms, in terms of the components of this system. Recall ([link]) is equivalent to

$$Y(z) = H(z)X(z)$$

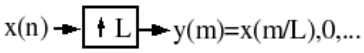
$$Y(\omega) = H(\omega)X(\omega)$$



and note that ([link]) is equivalent to

$$Y(z) = \sum_{m=-\infty}^{\infty} x(m)z^{-(Lm)} = x\left(z^L\right)$$

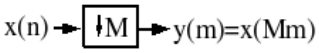
$$Y(\omega) = X(L\omega)$$



and ([link]) is equivalent to

$$Y(z) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{\frac{1}{M}}W_M^k\right)$$

$$Y(\omega) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(\frac{\omega}{M} + \frac{2\pi k}{M}\right)$$



$Y(z)$ is derived in the downsampler as follows:

$$Y(z) = \sum_{m=-\infty}^{\infty} x(Mm) z^{-m}$$

Let $n = Mm$ and $m = \frac{n}{M}$, then

$$Y(z) = \sum_{n=-\infty}^{\infty} x(n) \sum_{p=-\infty}^{\infty} \delta(n - Mp) z^{-\frac{n}{M}}$$

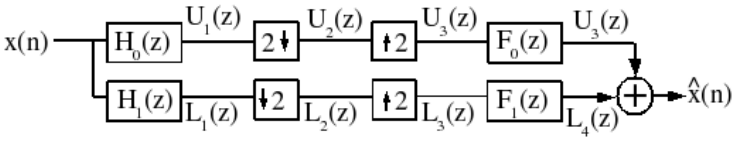
Now
Equation:

$$\begin{aligned} x(n) \textstyle\sum_{p=-\infty}^{\infty} \delta(n - Mp) &= \mathrm{IDFT}\left[x(\omega) * \frac{2\pi}{M} \sum_{k=0}^{M-1} \delta\left(\omega - \frac{2\pi k}{M}\right)\right] \\ &= \mathrm{IDFT}\left[\frac{2\pi}{M} \sum_{k=0}^{M-1} X\left(\omega - \frac{2\pi k}{M}\right)\right] \\ &= \frac{1}{M} \sum_{k=0}^{M-1} X(n) W_M^{-nk}\Big|_{W_M = e^{-\frac{i2\pi}{M}}} \end{aligned}$$

so
Equation:

$$\begin{aligned} Y(z) &= \textstyle\sum_{n=-\infty}^{\infty} \left(\frac{1}{M} \sum_{k=0}^{M-1} x(n) W_M^{-nk}\right) z^{-\frac{n}{M}} \\ &= \frac{1}{M} \sum_{k=0}^{M-1} x(n) \left(W_M^{+k} z^{\frac{1}{M}}\right)^{-n} \\ &= \frac{1}{M} \sum_{k=0}^{M-1} X\left(z^{\frac{1}{M}} W_M^{k}\right) \end{aligned}$$

Armed with these results, let's determine $\widehat{X}(z) \Leftrightarrow \widehat{x}(n)$. ([link])



Note

$$U_1(z) = X(z) H_0(z)$$

$$U_2(z) = \frac{1}{2} \sum_{k=0}^{1} X\left(z^{\frac{1}{2}} e^{-\frac{i2\pi k}{2}}\right) H_0\left(z^{\frac{1}{2}} e^{-(i\pi k)}\right) = \frac{1}{2} X\left(z^{\frac{1}{2}}\right) H_0\left(z^{\frac{1}{2}}\right) + \frac{1}{2} X\left(-z^{\frac{1}{2}}\right) H_0\left(-z^{\frac{1}{2}}\right)$$

$$U_3(z) = \frac{1}{2} X(z) H_0(z) + \frac{1}{2} X(-z) H_0(-z)$$

$$U_4(z) = \frac{1}{2} F_0(z) H_0(z) X(z) + \frac{1}{2} F_0(z) H_0(-z) X(-z)$$

and

$$L_4(z) = \frac{1}{2}F_1(z)H_1(z)X(z) + \frac{1}{2}F_1(z)H_1(-z)X(-z) = \frac{1}{2}F_1(z)H_1(z)X(z) + \frac{1}{2}F_1(z)H_1(-z)X(-z)$$

Finally then,
Equation:

$$\begin{aligned}\widehat{X}(z) &= U_4(z) + L_4(z) \\ &= \tfrac{1}{2}\left(H_0(z)F_0(z)X(z) + H_0(-z)F_0(z)X(-z) + H_1(z)F_1(z)X(z) + H_1(-z)F_1(z)X(-z)\right) \\ &= \tfrac{1}{2}\left(H_0(z)F_0(z) + H_1(z)F_1(z)\right)X(z) + \tfrac{1}{2}\left(H_0(-z)F_0(z) + H_1(-z)F_1(z)\right)X(-z)\end{aligned}$$

Note that the $X(-z) \rightarrow X(\omega + \pi)$ corresponds to the aliasing terms!

There are four things we would like to have:

1. No aliasing distortion
2. No phase distortion (overall linear phase → simple time delay)
3. No amplitude distortion
4. FIR filters

No aliasing distortion
By insisting that $H_0(-z)F_0(z) + H_1(-z)F_1(z) = 0$, the $X(-z)$ component of $\widehat{X}(z)$ can be removed, and all aliasing will be eliminated! There may be many choices for H_0, H_1, F_0, F_1 that eliminate aliasing, but most research has focused on the choice

$$F_0(z) = H_1(-z) : F_1(z) = -H_0(-z)$$

We will consider only this choice in the following discussion.

Phase distortion
The transfer function of the filter bank, with aliasing cancelled, becomes
$T(z) = \frac{1}{2}\left(H_0(z)F_0(z) + H_1(z)F_1(z)\right)$, which with the above choice becomes
$T(z) = \frac{1}{2}\left(H_0(z)H_1(-z) - H_1(z)H_0(-z)\right)$. We would like $T(z)$ to correspond to a linear-phase filter to eliminate phase distortion: Call

$$P(z) = H_0(z)H_1(-z)$$

Note that

$$T(z) = \frac{1}{2}\left(P(z) - P(-z)\right)$$

Note that $P(-z) \Leftrightarrow (-1)^n p(n)$, and that if $p(n)$ is a linear-phase filter, $(-1)^n p(n)$ is also (perhaps of the opposite symmetry). Also note that the sum of two linear-phase filters of the same symmetry (i.e., length of $p(n)$ must be **odd**) is also linear phase, so if $p(n)$ is an odd-length linear-phase filter, there will be no phase distortion. Also note that

$$Z^{-1}(p(z) - p(-z)) = p(n) - (-1)^n p(n) = \begin{cases} 2p(n) & \text{if n is odd} \\ 0 & \text{if n is even} \end{cases}$$

means $p(n) = 0$, when n is even. If we choose $h_0(n)$ and $h_1(n)$ to be linear phase, $p(n)$ will also be linear phase. Thus by choosing $h_0(n)$ and $h_1(n)$ to be FIR linear phase, we eliminate phase distortion and get FIR filters as well (condition 4).

Amplitude distortion
Assuming aliasing cancellation and elimination of phase distortion, we might also desire no amplitude distortion ($|T(\omega)| = 1$). All of these conditions require

$$T(z) = \frac{1}{2}\left(H_0(z)H_1(-z) - H_1(z)H_0(-z)\right) = cz^{-D}$$

where c is some constant and D is a linear phase delay. $c = 1$ for $|T(\omega)| = 1$. It can be shown by considering that the following can be satisfied!

$$T(z) = P(z) - P(-z) = 2cz^{-D} \Leftrightarrow \begin{cases} 2p(z) = 2c\delta(n-D) \ \text{ if } \ \text{n is odd} \\ p(n) = \text{anything } \ \text{ if } \ \text{n is even} \end{cases}$$

Thus we require

$$P(z) = \sum_{n=0}^{N'} p(2n)z^{-(2n)} + z^{-D}$$

Any factorization of a $P(z)$ of this form, $P(z) = A(z)B(z)$ can lead to a Perfect Reconstruction filter bank of the form

$$H_0(z) = A(z)$$

$$H_1(-z) = B(z)$$

[This result is attributed to Vetterli.] A well-known special case (Smith and Barnwell)

$$H_1(z) = -\left(z^{-(2D)+1}H_0\left(-z^{-1}\right)\right)$$

Design techniques exist for optimally choosing the coefficients of these filters, under all of these constraints.
Equation:

Quadrature Mirror Filters

$$H_1(z) = H_0(-z) \Leftrightarrow H_1(\omega) = H_0(\pi + \omega) = H_0^*(\pi - \omega)$$

for real-valued filters. The frequency response is "mirrored" around $\omega = \frac{\pi}{2}$. This choice leads to $T(z) = {H_0}^2(z) - {H_0}^2(-z)$: it can be shown that this can be a perfect reconstruction system only if

$$H_0(z) = c_0 z^{-(2n_0)} + c_1 z^{-(2n_1)}$$

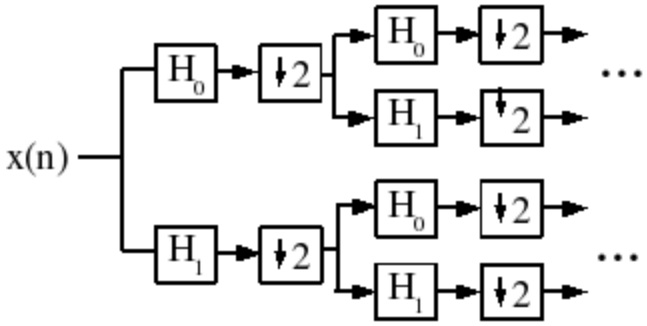
which isn't a very flexible choice of filters, and not a very good lowpass! The Smith and Barnwell approach is more commonly used today.

M-Channel Filter Banks

The theory of M-band QMFBs and PRFBs has been investigated recently. Some results are available.

Tree-structured filter banks

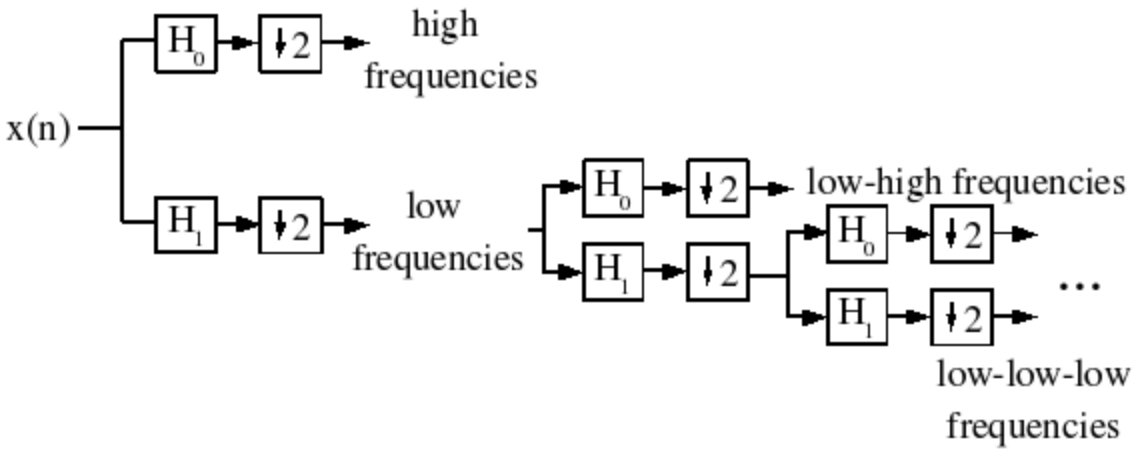
Once we have a two-band PRFB, we can continue to split the subbands with similar systems! ([link])



Thus we can recursively decompose a signal into 2 bands, each sampled at 2 th the rate of the original signal, and reconstruct exactly! Due to the tree structure, this can be quite efficient, and in fact close to the efficiency of an FFT filter bank, which does **not** have perfect reconstruction.

Wavelet decomposition

We need not split both the upper-frequency and lower-frequency bands identically. ([link])



This is good for image coding, because the energy tends to be distributed such that after a wavelet decomposition, each band has roughly equal energy.